UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,297	12/31/2003	Richard M. Shupak	MSFT-2568/307781.01	1690

41505 7590 05/11/2009
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
SCHMIDT, KARI L	

ART UNIT	PAPER NUMBER
2439	

MAIL DATE	DELIVERY MODE
05/11/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/750,297	**Applicant(s)** SHUPAK ET AL.	
	Examiner KARI L. SCHMIDT	**Art Unit** 2439	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 March 2009*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,6,19,20,23,28,37-40 and 45-51* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,6,19,20,23,28,37-40 and 45-51* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)☒ Claim(s) *10, 11, 15, 41, 43-44* are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 December 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 03/04/2009 has been entered.

Notice to Applicant

This communication is in response to the amendment filed on 03/04/2009. Claims 1, 2, 6, 10, 11, 15, 19, 20, 23, 28, 37-41 and 43-51 are currently pending for examination, however Claims 10, 11, 15, 41, 43, and 44 withdrawn from consideration as being directed to a non-elected invention. Claims 1, 5-6, 10, 14-15, 19, 28, 31, 33-34, and 36 have been amended. Claims 3-5, 7-9, 12-14, 16-18, 21-22, 24-27, 29-36, and 42 have been canceled.

Response to Arguments

Applicant's arguments with respect to claims 1, 2, 6, 10, 11, 15, 19, 20, 23, 28, 37-41 and 43-51 have been considered but are moot in view of the new ground(s) of rejection.

Election/Restrictions

Newly submitted claim 10 is directed to an invention that is independent or distinct from the invention originally claimed for the following reasons: the applicant has amended the claims to be based on "deriving a security cookie by XORing a secret value with each of the values retrieved from a jmp_buffer, the retrieved values precluding a first security cookie that has been stored previously in the jmp_buf buffer" and "comparing the derived security cookie against the first security cookie" and "if the derived security cookie matches the first security cookie then executing the runtime function" and "if the derived security cookie does not match the first security cookie then terminating execution of the runtime function" which differs from a target address which is compared to be found on the reference list of valid target addresses.

Since applicant has received an action on the merits for the originally presented invention, this invention has been constructively elected by original presentation for prosecution on the merits. Accordingly, claims 10, 11, 15, 41, 43, and 44 withdrawn from consideration as being directed to a non-elected invention. See 37 CFR 1.142(b) and MPEP § 821.03.

Claim Objections

Claim 51 objected to because of the following informalities: Claim 51 appears to be a run-on sentence and an incomplete claim. Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1, 2, 6, 28, 37-40, 50-51 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1

The examiner notes that “storing in a table, the list of valid target addresses as a reference list of valid target addresses” is indefinite. Where is this table stored? Is it in the object file or is in executable code or is it stored some where else (e.g. memory, etc). The examiner will interpret storing in a table of valid target addresses to be within an object file.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-2, 6, 28, 37-40 and 50-51 are rejected under 35 U.S.C. 101 s not falling within one of the four statutory categories of invention. While the claims recite a series of steps or acts to be performed, a statutory "process" under 35 U.S.C. 101 must (1) be tied to particular machine, or (2) transform underlying subject matter (such as an article or material) to a different state or thing. See page 10 of In Re Bilski 88 USPQ2d 1385. The instant claims are neither positively tied to a particular machine that accomplishes the claimed method steps nor transform underlying subject matter, and therefore do not qualify as a statutory process. The method including steps of compiling, storing, receiving, determining, and comparing is broad enough that the claim could be completely performed mentally, verbally or without a machine nor is any transformation apparent.

Claims 19-20, 23, 45-49 are rejected under 35 U.S.C. 101 s not falling within one of the four statutory categories of invention. Claims 19-20, 23, 45-49 are directed to "computer program products." Generally, functional descriptive material, such as a computer program, is statutory when it is stored on a tangible computer readable medium. See MPEP § 2106 IV.B.I(a). However, in the present application, the specification defines does not define "computer readable medium." A computer program listing on a sheet of paper is not considered to provide functionality, and is therefore considered to be merely **a computer program per se**, which is non-statutory subject matter. Further, "transmission media" such as "communications links" as broadly defined may include non-tangible media such as signals, which are also considered non-statutory. When a claim encompasses both statutory and non-statutory subject matter, the claim as a whole is directed to non-statutory subject matter.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 2, 6, 19, 23, 28, 37-40, and 45-51 are rejected under 35 U.S.C. 103(a), as best understood, as being unpatentable over Lueh (US 6,658,657 B1) in view of and Kaufer et al. (US 5,812,828) and Richarte, Gerardo. "Four different tricks to bypass StackShield and StackGuard protection".

Claims 1 and 19

Lueh discloses method, system and medium of processing runtime functions, comprising: compiling code to produce executable code (see at least, col. 5, lines 11-14: code is compiled for the first time before execution, which is the native code (e.g. executable code); receiving a call to a runtime function; determining associated data from the call to the runtime function; determining a target address from the associated data; comparing the target address with a reference list of valid target addresses stored in a table; and if the target address is found on the reference list of valid target addresses then executing the target (see at least, column 2, lines 9-45 : the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation. Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will by dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call… and column 4, lines 55-65).

Lueh fails to disclose **compiling code to produce executable code that is marked with an identifier indicating that the executable code comprises an object file containing a list of valid target addresses** for use in implementing runtime protection; **storing in a table, the list of valid target addresses as a reference list of valid target address**; receiving a call to a runtime function of the executable code for a runtime function; and if the target address is not found on the reference list of a valid target addresses then terminating execution of the runtime function.

Kaufer discloses **compiling disclose compiling code to produce executable code that is marked with an identifier indicating that the executable code comprises an object file containing a list of valid target addresses** (see at least, col. 3, lines 57-col. 4, lines 42: the examiner notes that an executable file a.out is linked with object files foo.o and bar.o and further the linking of the compiler will associate (e.g. identify) the object files associated with a.out (e.g. conventional compiling technique known in the arts) further the examiner notesthe these object files contains one or more instructions that are called during execution of the a.out) and **storing in a table, the list**

of valid target addresses as a reference list of valid target address (see at least, col. 3, lines 57-col. 5, lines 419: the examiner notes that during the simulation run a table is built that includes instructions for the checking of address space for the given instruction).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh to include compiling code to produce executable code that is marked with an identifier indicating that the executable code comprises an object file containing a list of valid target addresses and storing in a table, the list of valid target addresses as a reference list of valid target address as taught by Kaufer. One of ordinary skill in the art would have been motivated to combine the teachings in order to provide the checking of instructions for each function within the computer readable code (see at least, Kaufer, col. 1, lines 59-64).

Lueh in view of Kaufer fails to disclose code to produce executable code that is marked with an identifier indicating that the executable code supports runtime protection; receiving a call to a runtime function of the executable code for a runtime function; and if the target address is not found on the reference list of a valid target addresses then terminating execution of the runtime function

Richarte discloses compiling code to produce executable code that is marked with an identifier indicating that the executable code supports runtime protection (see at least, page 4, 2.2 StackGuard Protection: the examiner notes an output file from the gcc operation is executable code which is marked with a canary (e.g. identifier) for runtime protection)); receiving a call to a runtime function of the executable code for a runtime function (see at least, page 4, 2.2. StackGuard Protection: the examiner notes function prologue and epilogue would be contained in a programs body); and if the target address is not found on the reference list of a valid target addresses then terminating execution of the executable code (see at least, 2.3 StackSheild Protection (2.3.2 Checked Clones): the examiner notes the comparison of addresses on a stack and if not matched a SYS_exit system call is placed (e.g. termination of the executable code)).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh in view of Kaufer to include compiling code to produce executable code that is marked with an identifier indicating that the executable code supports runtime protection; receiving a call to a runtime function of the executable code for a runtime function; and if the target address is not found on the reference list of a valid target addresses then terminating execution of the executable code as taught by Richarte. One of ordinary skill in the art would have been motivated to combine the teachings in order to provide stack shielding technologies to protect programs against exploitations of stack based buffer overflows (see at least, Richarte, Abstract).

Claims 2 and 20

Lueh discloses the method of claim 1, wherein the step of determining the associated data comprises accessing data in a data structure connected with the runtime function and calculating the associated data based on the accessed data (see at least, column 1, lines 12-65, Figures 1 and 2, column 4, lines 45--65).

Claims 6 and 23

Lueh discloses the method of claim 1 comprising the step of generating the reference list of valid target addresses during execution of a previous runtime function (see at least, column 2, lines 46-64: if A's target address of foo is compiled and ran an address is now known, and therefore the JIT address of A's foo will be used (e.g. occur after a first compilation with A's foo without a known address)).

Claims 28

Lueh discloses the method of claim 1 comprising the step of storing the target address in a caller provided location during execution of a previous runtime function (see at least, column 2, lines 9-45: a vtable is a provided location and if A's target address of foo is compiled and ran an address is now known, and therefore the JIT address of A's foo will be used (e.g. occur after a first compilation with A's foo without a known address).

Claims 37 and 45

Lueh discloses determining if at least a portion of the associated data is valid (see at least, column 2, lines 9-45 : the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation. Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will by dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call… and column 4, lines 55-65).

Lueh in view of Kaufer fails to disclose preventing execution of the target if the associated data is not valid.

Richarte discloses disclose preventing execution of the target if the associated data is not valid (see at least,.3 StackSheild Protection (2.3.2 Checked Clones): comparison of addresses on a stack and if not matched a SYS_exit system call is placed (e.g. termination of the executable code)).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh in view of Kaufer to include preventing execution of the target if the associated data is not valid as taught by Richarte. One of ordinary skill in the art would have been motivated to combine the teachings in order to provide stack shielding technologies to protect programs against exploitations of stack based buffer overflows (see at least, Richarte, Abstract).

Claims 38 and 47

Lueh in view of Kaufer fails to disclose wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies.

Richarte discloses to wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies (see at least, 2.2. StackGuard Protection: they use pushing and comparing canary (e.g. a secruiyt cookie) on a stack and 2.3 StackSheild Protection (2.3.2 Checked Clones): comparison of addresses on a stack and if not matched a SYS_exit system call is placed (e.g. termination of the executable code)).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh in view of Kaufer to include wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies as taught by Richarte. One of ordinary skill in the art would have been motivated to combine the teachings in order to provide stack shielding technologies to protect programs against exploitations of stack based buffer overflows (see at least, Richarte, Abstract).

Claims 39 and 48

Lueh discloses further comprising determining and storing a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function (see at least, col. 2, lines 46-64: the examiner notes an allocation of memory space is a predetermined calculated value based on the function prior to the runtime call).

Claims 40 and 49

Lueh discloses wherein determining if the associated data is valid comprises comparing the predetermined calculated value to another calculated value based on the associated data (see at least, col. 2, lines 9-45: the examiner notes the checking and comparing of the vtable).

Claim 46

Lueh discloses further comprising a storage device that stores a list of valid targets, wherein the dispatcher system determines if the associated data is valid by comparing the target address to the list of valid target addresses (see at least, col. 2, lines 9-45: the examiner notes the checking and comparing of the vtable and col. 4, lines 56-col. 4, line 20: the examiner notes ROM, RAM, disk storage mediums, and etc.).

Claim 50

Lueh in view of Kaufer and Ricarte discloses the use of an linking executable code to object fiels (see Kaufer, col. 3, lines 57-col. 4, lines 42) and further an identifier that is 4 bytes long that indicates that the executable code implements runtime protections (see Ricarte, page 5).

Lueh in view of Kaufer and Ricarte fails to disclose wherein the identifier is a bit.

The examiner notes it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh in view of Kaufer and Ricarte to modify the 4 byte long identifier and design it to be a bit as a matter of design choice. The examiner notes that a programmer with knowledge of one or ordinary skill in the art would have had the knowledge to modify bytes into bits as a matter of conserving space within memory.

Claim 51

Lueh in view of Kaufer and Ricarte discloses the use a table which links to the call of a runtime function (see Lueh, col. 2, lines 9-45).

Lueh in view of Kaufer and Ricarte fails to disclose wherein the table is a .setjmp table.

The examiner notes it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh in view of Kaufer and Ricarte to modify the vtable’s name to be a “.setjmp” table which is a matter design choice. The examiner notes that a programmer with knowledge of one or ordinary skill in the art would have had the knowledge to modify names of tables for purposes of understanding what a table does with respect to code execution.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KARI L. SCHMIDT whose telephone number is (571) 270-1385. The examiner can normally be reached on Monday - Friday: 7:30am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kambiz Zand can be reached on 571-272-3811. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Kari L Schmidt/
Examiner, Art Unit 2439

/Kambiz Zand/
Supervisory Patent Examiner, Art Unit 2434